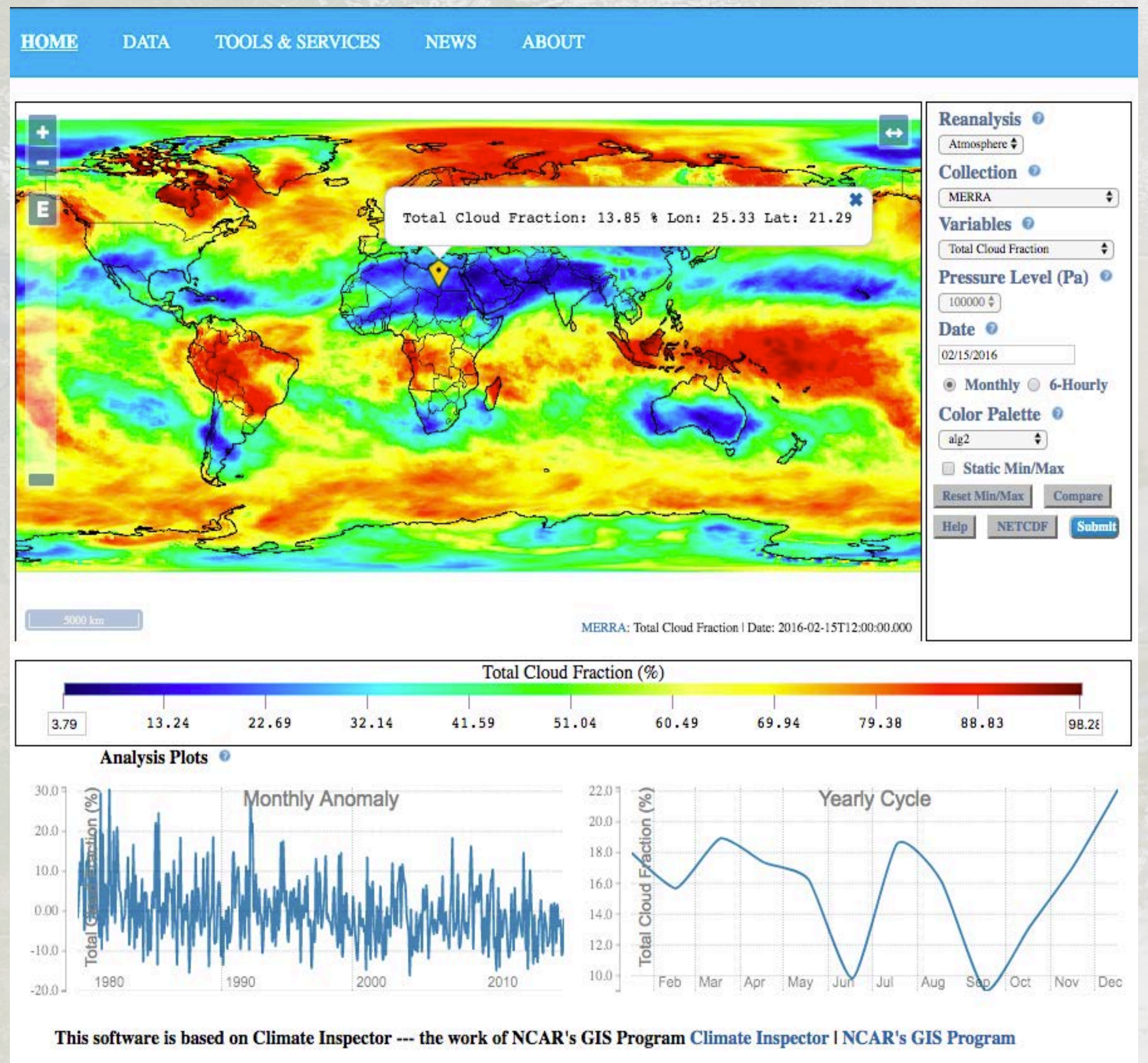
NCCS
NASA CENTER FOR CLIMATE SIMULATION
HIGH-PERFORMANCE SCIENCE

The Earth Data Analytic Services (EDAS) Framework

Thomas Maxwell, Dan Duffy, Laura Carriere, Jerry Potter
NASA GODDARD SPACE FLIGHT CENTER, GREENBELT, MD

NASA

CREATE-V Application



Total Cloud Fraction: 13.85 % Lon: 25.33 Lat: 21.29

Reanalysis: Atmosphere
Collection: MERRA
Variables: Total Cloud Fraction
Pressure Level (Pa): 100000
Date: 02/15/2016
● Monthly ○ 6-Hourly
Color Palette: alg2
□ Static Min/Max

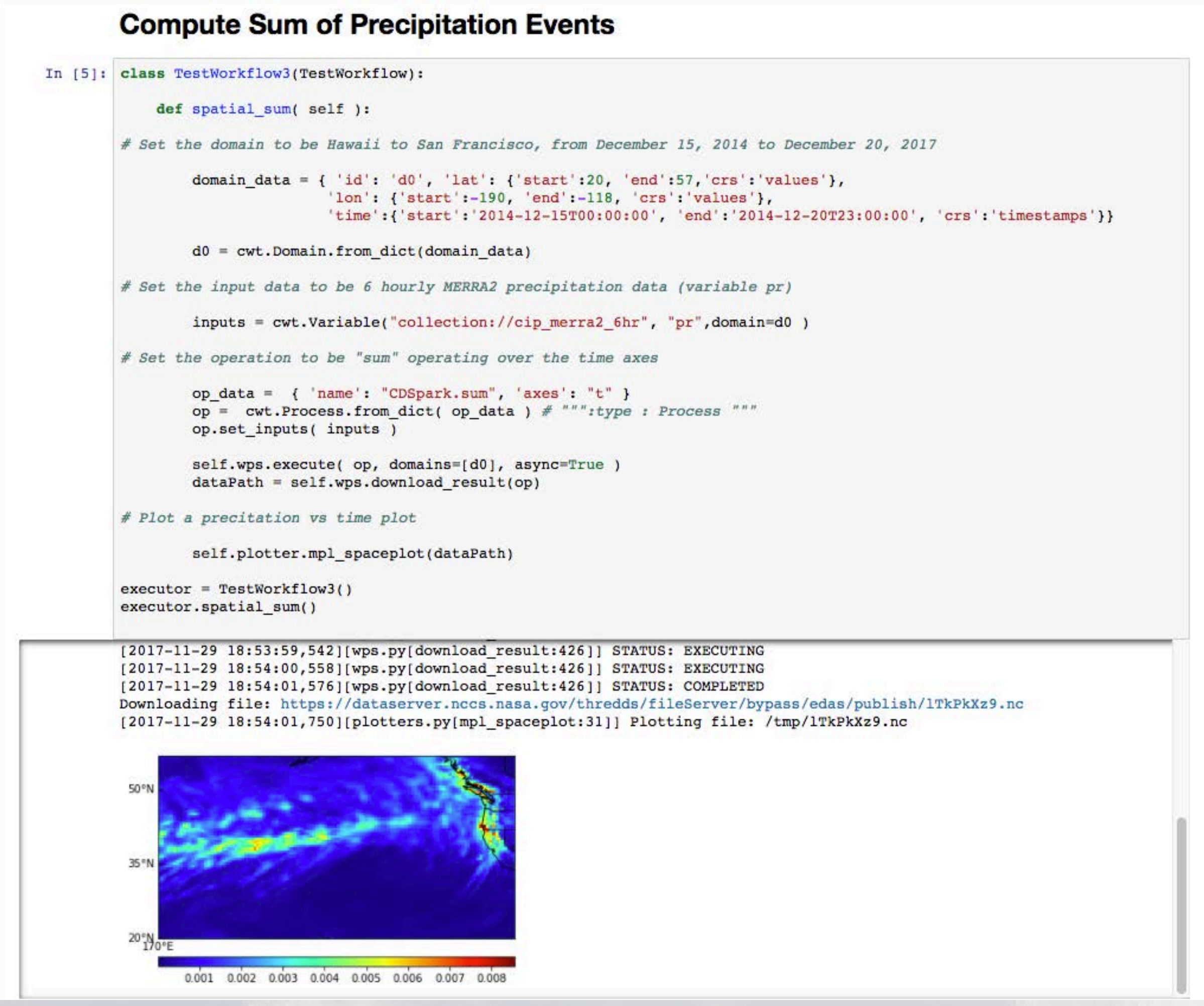
MERRA: Total Cloud Fraction | Date: 2016-02-15T12:00:00.000

This software is based on Climate Inspector --- the work of NCAR's GIS Program Climate Inspector | NCAR's GIS Program

This interactive web application expands GIS mapping capabilities for manual analysis of reanalysis data. It harnesses EDAS analysis services to interactively generate departures and annual cycles for time series plots in response to a user's mouse click on a map.

Poster IN21D-0064: *CREATE-IP and CREATE-V: Data and Services Update*.

Jupyter Notebook Interface



Compute Sum of Precipitation Events

```
In [5]: class TestWorkflow3(TestWorkflow):

            def spatial_sum( self ):

        # Set the domain to be Hawaii to San Francisco, from December 15, 2014 to December 20, 2017

                domain_data = { 'id': 'd0', 'lat': {'start':20, 'end':57,'crs':'values'},
                                'lon': {'start':-190, 'end':-118, 'crs':'values'},
                                'time':{'start':'2014-12-15T00:00:00', 'end':'2014-12-20T23:00:00', 'crs':'timestamps'}}

                d0 = cwt.Domain.from_dict(domain_data)

        # Set the input data to be 6 hourly MERRA2 precipitation data (variable pr)

                inputs = cwt.Variable("collection://cip_merra2_6hr", "pr",domain=d0 )

        # Set the operation to be "sum" operating over the time axes

                op_data =  { 'name': "CDSpark.sum", 'axes': "t" }
                op =  cwt.Process.from_dict( op_data ) # """:type : Process """
                op.set_inputs( inputs )

                self.wps.execute( op, domains=[d0], async=True )
                dataPath = self.wps.download_result(op)

        # Plot a precitation vs time plot

                self.plotter.mpl_spaceplot(dataPath)

        executor = TestWorkflow3()
        executor.spatial_sum()
```

```
[2017-11-29 18:53:59,542][wps.py[download_result:426]] STATUS: EXECUTING
[2017-11-29 18:54:00,558][wps.py[download_result:426]] STATUS: EXECUTING
[2017-11-29 18:54:01,576][wps.py[download_result:426]] STATUS: COMPLETED
Downloading file: https://dataserver.nccs.nasa.gov/thredds/fileServer/bypass/edas/publish/lTkPkXz9.nc
[2017-11-29 18:54:01,750][plotters.py[mpl_spaceplot:31]] Plotting file: /tmp/lTkPkXz9.nc
```

The EDAS Python API is used to create and manage analysis workflows. EDAS Python scripts can be executed standalone or within Jupyter notebook cells. The API facilitates notebook-embedded plotting of analysis results. A collection of sample Jupyter notebooks can be found at: *https://www.nccs.nasa.gov/services/Analytics*

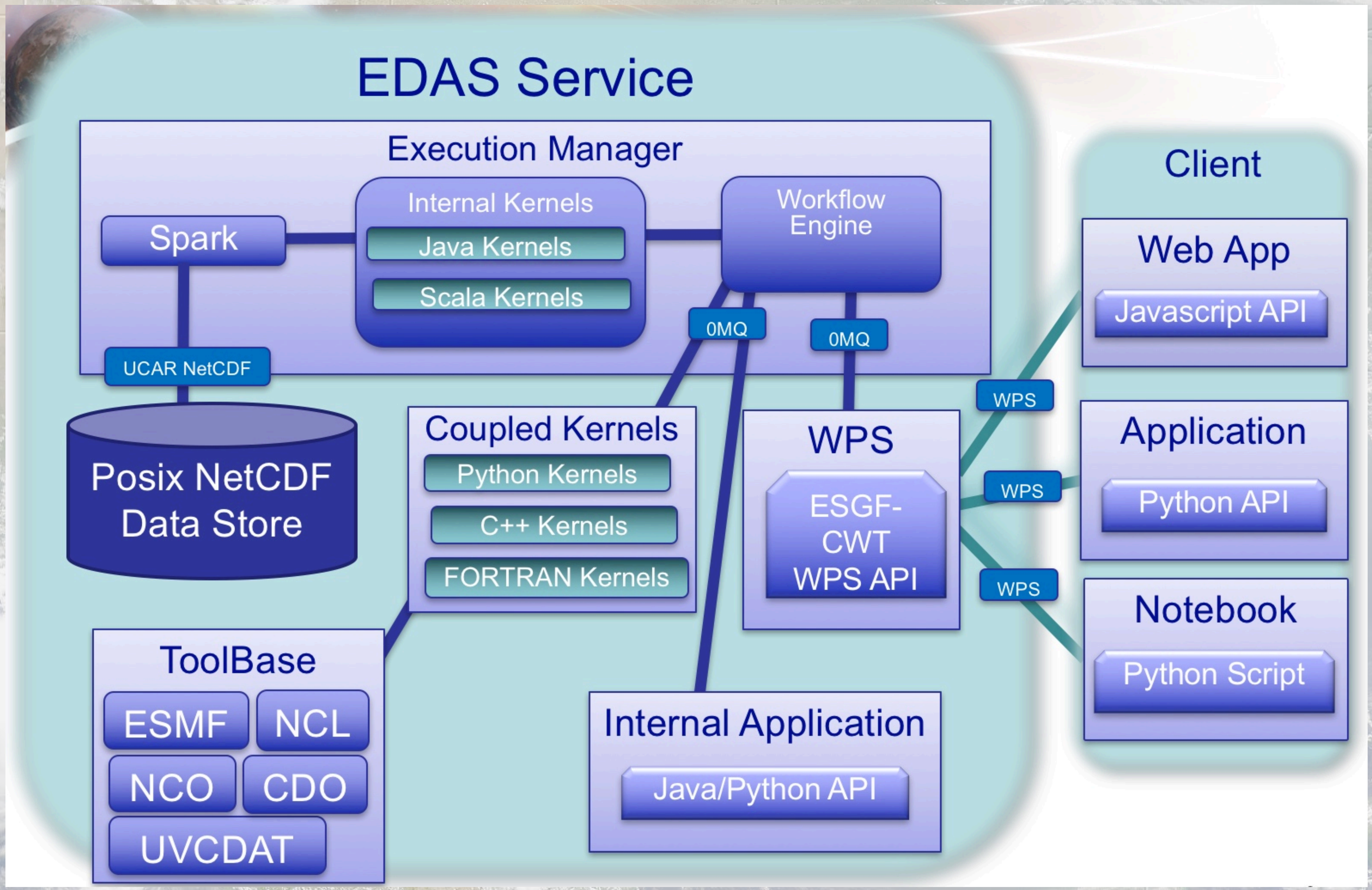
ABSTRACT

Faced with unprecedented growth in Earth data volume and demand, NASA has developed the Earth Data Analytic Services (EDAS) framework, a high-performance big data analytics framework built on Apache Spark. This framework enables scientists to execute data processing workflows combining common analysis operations close to the massive data stores at NASA. The data is accessed in standard (NetCDF, HDF, etc.) formats in a POSIX file system and processed using vetted Earth data analysis tools (ESMF, CDAT, NCO, etc.). EDAS utilizes a dynamic caching architecture, a custom distributed array framework, and a streaming parallel in-memory workflow for efficiently processing huge datasets within limited memory spaces with interactive response times.

EDAS services are accessed via a WPS API being developed in collaboration with the ESGF Compute Working Team to support server-side analytics for ESGF. New analytic operations can be developed in Python, Java, or Scala (with support for other languages planned). Client packages in Python, Java/Scala, or JavaScript contain everything needed to build, submit, manage, and visualize big data analysis workflows from the user's desktop computer or to develop web applications with embedded analytics.

The EDAS architecture brings together the tools, data storage, and high-performance computing required for timely analysis of large-scale data sets, where the data resides. It is currently deployed at NASA and available for public use. Another NASA EDAS deployment supports the Collaborative REAnalysis Technical Environment (CREATE) project, which centralizes numerous global reanalysis datasets onto a single analytics platform. These services enable scientists and decision makers to access remote model/reanalysis data archives and investigate trends, variability, anomalies, and other features of local and global earth system dynamics.

EDAS: Analysis-as-a-Service Infrastructure (AaaS)


EDAS Service

Execution Manager

Spark

Internal Kernels

Java Kernels

Scala Kernels

Workflow Engine

UCAR NetCDF

0MQ

0MQ

Posix NetCDF Data Store

Coupled Kernels

Python Kernels

C++ Kernels

FORTRAN Kernels

WPS

ESGF-CWT WPS API

WPS

WPS

WPS

Client

Web App

Javascript API

Application

Python API

Notebook

Python Script

ToolBase

ESMF

NCL

NCO

CDO

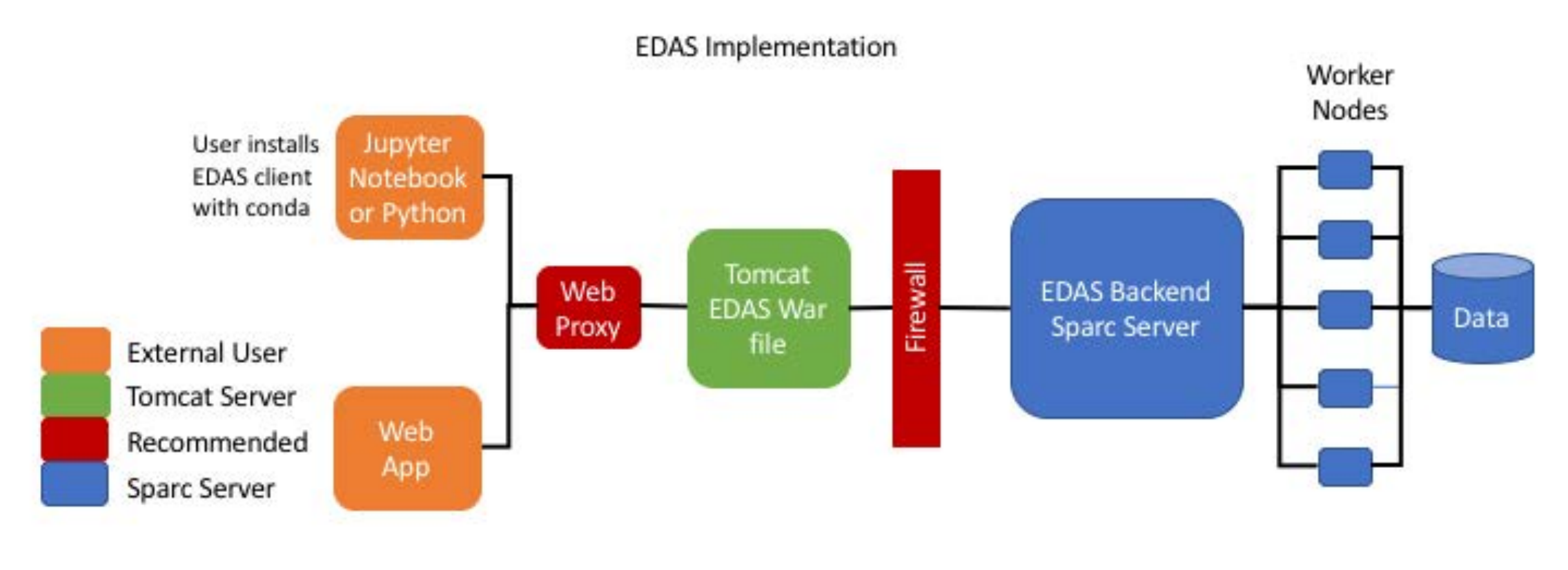
UVCDAT

Internal Application

Java/Python API

EDAS Deployment

- **NASA EDAS deployment on the DASS:**
  - Data Analytics and Storage System (DASS): *https://www.nccs.nasa.gov/services/dass*
  - Web portal (WPS server): *https://edas.nccs.nasa.gov/wps/cwt*
  - Available data collections:
    - *https://edas.nccs.nasa.gov/wps/cwt?request=GetCapabilities&identifier=coll*
- **EDAS Distribution:**
  - Server: *https://github.com/nasa-nccs-cds/EDAS.git*
  - Web app: *https://github.com/nasa-nccs-cds/CDWPS.git*
  - Client: *https://github.com/ESGF/esgf-compute-api.git*
- **Documentation:**
  - *https://www.nccs.nasa.gov/services/Analytics*


EDAS Implementation

User installs EDAS client with conda

Jupyter Notebook or Python

Web App

Web Proxy

Tomcat EDAS War file

Firewall

EDAS Backend Sparc Server

Worker Nodes

Data

🟧 External User
🟩 Tomcat Server
🟥 Recommended
🟦 Sparc Server

EDAS Canonical Operators

- **Reduce Operations:**
  - Max, Min, Sum, Average, RootMeanSquare
- **Combine (Ensemble) Operations:**
  - Max, Min, Sum, Average, Difference, Multiply, Divide
- **Common Workflows:**
  - Anomaly (Ave+Diff), StdDev (Ave+Diff+ RMS), etc.
- **Utility Operations :**
  - Subset, Regrid, Filter
- **Current Operator List:**
  - https://edas.nccs.nasa.gov/wps/cwt?request=GetCapabilities

Why is this approach distinctive?

- **Direct access to NetCDF data archives via disk or OpenDAP:**
  - *Alleviates the need to maintain additional copies of the data.*
- **Deploys existing (Python) climate data analysis tools:**
  - *Utilizes UVCDAT, ESMF, and other Python analytic toolkits.*
  - *No changes to existing applications.*
  - *Parallelizes the data, not the applications.*
- **High-performance analytics:**
  - *Optimizes decomposition over processors for the current task.*
  - *Streaming in=memory parallel workflows using Apache Spark.*
  - *Parallel GPFS data access (faster then HDFS on our cluster).*
  - *15 to 50 times faster then standard tools in our environment.*
- **Modular structure:**
  - *Easily add new technologies and compare approaches.*
  - *Deploy optimal solution for problem domain.*
  - *Build new workflows by composing canonical operations.*
- **ESGF CWT WPS API:**
  - *Designed to operate as a compute engine for the ESGF.*